Request Reliability Augmentation With Service Function Chain Requirements in Mobile Edge Computing

Weifa Liang, *Senior Member, IEEE*, Yu Ma, Wenzheng Xu, *Member, IEEE*, Zichuan Xu, *Member, IEEE*, Xiaohua Jia, *Fellow, IEEE*, and Wanlei Zhou, *Senior Member, IEEE*

Abstract—Provisioning reliable network services for mobile users in edge computing environments is the top priority of network service providers, as unreliable services will result in tremendous losses of revenues and customers. In this paper, we study a novel service reliability augmentation problem in a mobile edge computing (MEC) network, where mobile users request network services with service function chain (SFC) and reliability expectation requirements. To enhance the service reliability of user requests, it is a common practice to make use of redundant virtualized network function (VNF) instance placement in case the primary VNF instance fails. We aim to augment the service reliability of each admitted request to its specified reliability expectation, subject to computing capacity on each cloudlet. To this end, we first formulate a novel service reliability augmentation problem for each request with an SFC and a reliability expectation requirement, by augmenting its reliability through redundant VNF instance deployment. We then show that the problem is NP-hard, and provide an admission framework of user requests by placing primary VNF instances of network functions in the SFC to different cloudlets. We then deal with the service reliability augmentation problem of an admitted request under the assumption that all secondary VNF instances of each primary VNF instance must be placed into the cloudlets no more than l hops from the cloudlet of its primary VNF instance for a fixed l with $1 \leq l \leq n-1$, where n is the number of cloudlets in the network, for which we formulate an integer linear program solution, and develop a randomized algorithm with a good approximation ratio and high probability, at the expense of moderate resource constraint violations. We also devise a deterministic heuristic for the problem without any resource violation. We third study the service reliability augmentation problem for a set of admitted requests by extending the proposed algorithm for the service reliability augmentation problem for a single request admission. We finally evaluate the performance of the proposed algorithms through experimental simulations. Experimental results demonstrate that the proposed algorithms are promising, and their empirical results are superior to their analytical counterparts.

Index Terms—Reliability augmentation of services, virtualized network function (VNF) placement, primary and secondary VNF instance placement, approximation algorithms, budgeted minimum cost generalized assignment problems, Mobile Edge Computing (MEC), l-hop message communication model, randomized algorithms, VNF instance redundancy

1 Introduction

Network Function Virtualization (NFV) and Mobile Edge Computing (MEC) have been envisioned as key enabling technologies to support delay-sensitive applications in smart cities, the Internet of Things (IoTs), and intelligent transportation. NFV decouples network functions (NFs) from dedicated hardware - middleboxes, leading to significant cost reduction in network service provisioning. Network service providers provide mobile users with low-latency, highly reliable network services through the placement of Virtual Network Functions (VNFs) to cloudlets in a mobile edge-cloud network to meet user service demands with service function chain (SFC) and reliability expectation requirements. Due to the chaining nature and distributed placement of VNF instances, the failure of any single VNF instance in a chain will heavily affect the normal operation of a service, thereby resulting in serious data loss and resource waste. To enhance the service reliability, the redundant backup is a common practice to improve service reliability, i.e., the redundant placement of VNF instances to cloudlets as backups.

In this paper, we consider reliability-aware network service provisioning in an MEC environment, where each mobile user requests service with a service function chain (SFC) and a reliability expectation requirements. To improve user experience on the use of virtualized services while meeting their reliability expectations ultimately, the deployment of redundant VNF instances is a common choice, by placing multiple redundant VNF instances for each network function in the SFC to different cloudlets. We distinguish between a single primary VNF instance and multiple secondary VNF instances for each

- *Weifa Liang and Xiaohua Jia are with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. E-mail: {weifa.liang, csjia}@cityu.edu.hk.*
- *Yu Ma is with the Research School of Computer Science, The Australian National University, Canberra, ACT 2601, Australia. E-mail: yu.ma@anu.edu.au.*
- *Wenzheng Xu is with the School of Computer Science, Sichuan University, Chengdu 510006, China. E-mail: wenzheng.xu@scu.edu.cn.*
- *Zichuan Xu is with the School of Software, Dalian University of Technology, Dalian 116621, China. E-mail: z.xu@dult.edu.cn.*
- *Wanlei Zhou is with the Institute of Data Science, City University of Macau, Macao SAR 999078, China. E-mail: wlZhou@cityu.mo.*

Manuscript received 1 December 2020; revised 1 April 2021; accepted 16 May 2021. Date of publication 18 May 2021; date of current version 3 November 2022.
(Corresponding author: Weifa Liang.)
Digital Object Identifier no. 10.1109/TMC.2021.3081681

network function [10]: the former is an active VNF instance while the latter are idle ones until the primary one fails. Considering limited resources in an MEC network, how to augment the service reliability of an admitted request poses great challenges. For example, how many secondary VNF instances of each primary VNF instance in the SFC of a request are needed? and to which cloudlets will these secondary VNF instances be placed if all secondary VNF instances of a primary VNF instance must be placed in the cloudlets no more than l-hops from the cloudlet of its primary VNF instance with a fixed $l \geq 1$? Due to the non-linearity of the optimization objective of the problem, how to find an efficient solution to this non-linear optimization problem? Furthermore, the service provider of an MEC network provides its computing and storage resources for various user services, which implies a bunch of users will compete for the limited resources. As different user requests have different service function chains, how to allocate limited resources to meet the service demands of a set of users while meeting SFC reliability requirements?

The novelties of the work in this paper lie in the formulation of a novel service reliability augmentation problem for each admitted request with a service function chain and a given reliability expectation requirements, under a l-hop communication model that all secondary VNF instances of any network function must be placed into the cloudlets no more than l-hops from the cloudlet hosting its primary VNF instance with $1 \leq l \leq n-1$, where n is the number of cloudlets in an MEC network. Due to nonlinearity of the optimization objective, we devise a randomized algorithm with high probability and an efficient heuristic algorithm for the problem. We also extend the problem solution to solve the service reliability augmentation problem for a set of admitted requests.

The main contributions of this paper are presented as follows. We first formulate a novel service reliability augmentation problem for an admitted request with an SFC and a reliability expectation requirements in an MEC network, and show that the problem is NP-hard. We then propose an admission framework of user requests through the placement of primary VNF instances of network functions in the SFC of each request into different cloudlets, under the assumption that all secondary VNF instances of each primary VNF instance must be placed into the cloudlets no more than l hops from the cloudlet of their primary VNF instances, where l is fixed with $1 \leq l \leq n-1$, and the value of l is used to control the latency of updating its secondary VNF states if there is any update on a primary VNF instance. We third formulate a non-trivial integer linear program (ILP) solution to the service reliability augmentation problem through reducing the problem to another optimization problem, and develop a randomized algorithm with high probability for it at the expense of moderate (bounded) resource constraint violations. We also propose a deterministic algorithm through reducing the problem to a series of minimum-cost maximum matching problems without any resource constraint violations. We fourth devise an algorithm for the service reliability augmentation problem for a set of admitted requests through invoking the proposed algorithms for the service reliability augmentation problem for a single admitted request. We finally evaluate the performance of the proposed algorithms through experimental simulations. Experimental results demonstrate that the proposed algorithms are promising and outperform their analytical counterparts.

The rest of the paper is organized as follows. Section 2 summarizes the related work of reliable service function provisioning. Section 3 introduces notions, notations, and the problem definitions. The NP-hardness of the problems is also shown in this section. Section 4 proposes a framework of admitting a request with an SFC and a given reliability expectation requirement. Section 5 formulates an integer linear program (ILP) solution for the service reliability augmentation problem for an admitted request, under the assumption that all the secondary VNF instances of each primary VNF instance must be placed within l-hop cloudlets from the cloudlet of the primary VNF instance for a fixed integer l with $1 \leq l \leq n-1$. Section 6 develops a randomized approximation algorithm based on the linear relaxation of the ILP solution. Section 7 devises an efficient heuristic algorithm for the problem. Section 8 proposes an algorithm for the service reliability augmentation problem for a set of admitted requests. Section 9 evaluates the proposed algorithms empirically, and Section 10 concludes the paper.

2 Related Work

As a key-enabling technology of 5G and the next generation 6G networks, MEC has gained tremendous attentions in the research community recently [21]. There are extensive studies on virtualized network service provisioning in MEC [5], [14], [26], [28]. For example, Rodriguez-Santana *et al.* [26] designed a task offloading framework for augmented reality applications on mobile devices. They assumed that users can share VM instances and assign arrived user requests to existing VM instances, or creating new instances, while satisfying their delay requirements. Xu *et al.* [28] studied the problem of maximizing the network throughput while minimizing the cost of request admissions, and developed an effective prediction mechanism to create or release VNF instances of different network functions for cost savings. Feng *et al.* [5] proposed an algorithm with performance guarantee for placing VNFs in distributed cloud networks and routing service flows among the placed VNFs under the constraints of the service function chains of requests. Their solution is achieved through a reduction to reduce the problem to a multi-commodity-chain flow problem on a cloud-augmented graph.

There are intensive efforts on the reliability (or availability) of virtualized service function provisioning in data-center networks and MEC networks in the past several years, and a recent survey on this topic is given by Han *et al.* [10]. For example, Chemodanov *et al.* [3] studied a reliable service function chain composition problem which can deal with both SFC demand fluctuations and infrastructure outage uncertainties for geo-distributed data centers. They proposed the problem as an integer multi-commodity-chain flow problem and solved it by adopting a metapath-based composite variable approach. Shang *et al.* [27] considered VNF backups to minimize the cost while meeting the service function chain availability requirements in an online manner. It uses both static backups and dynamic ones created on the fly to accommodate the resource limitation of edge networks. Their approach does not assume failure rates of VNFs but instead strives to find a tradeoff between the desired availability of SFCs and the backup cost. Yu *et al.* [30] explored a QoS-aware and reliable traffic steering problem in mobile networks, considering heterogeneous requirements, including QoS (throughput and delay), reliability, security and type-of-transmission constraints. They proved the problem is NP-hard, and developed a fully-polynomial time approximation scheme (FPTAS) for the problem. Fan *et. al* [7], [8] studied the availability issue of service function chains. They proposed heuristic algorithms that map SFCs to servers in data center networks with the aim of minimizing the numbers of on-site and off-site backups required, in order to meet the given availability requirements. Fan *et al.* [9] considered the reliable-SFC instance service providing in a data center network, where there are sufficient computing resource for VNF instance placement, and all VNF instances (both primary and secondary VNF instances) of a function service chain is consolidated into a single server. Qu *et. al* [23] jointly considered the availability and delay constraints in the backup resource allocation problem of SFC with an objective to minimize the amount of bandwidth resource needed in a data center network, and they later extended their work by allowing the sharing of VNF instances in [24]. Ding *et al.* [4] formulated how to calculate VNF placement availability when at most one backup chain is allowed. They proposed a heuristic to find the backup SFCs with the minimum cost such that the accumulative availability for each request is met. Aidi *et al.* [1] proposed a framework to efficiently manage survivability of service function chains and the backup VNFs, with the aim to determine both the minimum number and optimal locations of backup VNFs to protect service function chains. They proposed heuristics for the problem. Yang *et al.* [29] studied the delay-sensitive and availability-aware NFV scheduling problem, which takes the NFV placement availability constraint into account, by proposing a model quantitatively calculate the traversing delay for a flow in an SFC, and proposed an integer nonlinear programming and an efficient heuristic for the problem.

There are also several studies focusing on the robust service provisioning in MEC, where each request has only a single service function rather than a service function chain requirement and with or without delay constraints. For example, Huang *et al.* [13] studied the robust network function service provisioning in mobile edge computing environments, for which they developed two provable approximation algorithms for primary and secondary VNF instance placements among cloudlets in an MEC network, assuming that each service chain contains only one network function. Under an ideal assumption that both network function and server failure probabilities are given, and backup VNF instances of any function should be placed into the same server, and all different functions have the same computing resource demands, He *et al.* [11] considered the assignment of backup VNF instances to different servers such that the maximum failure probability of the functions is minimized, and they provided two heuristic algorithms for the problem. Li *et al.* [15], [16] investigated the VNF instance placement of dynamic requests with a single VNF request with the aim to meet individual requests' reliability requirements. Under the assumption that requests arrive into the system dynamically without the knowledge of future arrivals, they devised an online algorithm with a constant competitive ratio for the problem when all VNF instances of the VNF of an admitted request are consolidated into a single cloudlet, the approximate solution obtained is at the expense of moderate bounded resource violations. Li *et al.* [17] recently studied the robust service function chain placement (RSFCP) problem with the aim to maximize the expected profit of the service provider, for which they devised a Markov-chain based approximation algorithm by admitting as many requests as possible while meeting the latency requirement of each admitted request. However, all of the mentioned studies focused on requests with a single VNF service, not a VNF service function chain, and none of the studies has ever considered the service reliability augmentation issue on admitted requests through the placement of extra VNF instances to different cloudlets. Lin *et al.* [19] recently studied the primary and backup VNF instance placement for a service chain in an MEC network to meet the specified reliability requirement of a request, for which they proposed a randomized algorithm and a heuristic algorithm, assuming that the primary and backup VNF instances can be placed to any cloudlets as long as there are sufficient computing resources in the cloudlets to accommodate the VNF instances.

Unlike the aforementioned studies that either conducted in datacenter networks or MEC networks, in this paper we study the provisioning of reliable services through enhancing service reliability. We focus on the service reliability augmentation problem for an admitted request or a set of admitted requests through redundant VNF instance placements of different network functions in its SFC to different cloudlets, subject to the computing capacity on each cloudlet, under the assumption that all redundant VNF instances of a network function must be placed to the cloudlets no more than l-hops from the cloudlet of its primary VNF instance with $l = 1, \dots, n-1$. It must be mentioned that this paper is an extension of a conference paper [18].

3 System Model

Consider that the MEC network is an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of links between nodes. Each node $v \in V$ is an Access Point (AP), which may or may not be co-located with a cloudlet. If it is co-located with a cloudlet v, the computing capacity of the cloudlet is $C_v > 0$, otherwise, its computing capacity $C_v = 0$. Let $N_l(v)$ be the l-neighbor set of cloudlet v in G, where $N_l(v) = \{u \mid \text{the distance between } u \text{ and } v \text{ in terms of the number of } hopsisnogreaterthanl \text{ in } G\}$ with a fixed l and $1 \leq l \leq |V| - 1$. Denote by $N_l^+(v) = N_l(v) \cup \{v\}$, e.g., $N_1^+(v) = \{u \mid (u, v) \in E\} \cup \{v\}$ when $l = 1$. Let $\mathcal{F}$ be the set of all different network functions offered by the system.

3.1 Request Admission and its Reliability Utility Function

Let SFC_j of request j consist of L_j different network functions $f_1, f_2, \dots, f_{L_j}$ in order and has a reliability expectation ρ_j, where the reliability of any VNF instance of network function f_i is a given value r_i with $0 < r_i \leq 1$. Assuming that request j has been admitted, all primary VNF instances of its SFC_j have been placed, and its reliability achieved $\Pi_{i=1}^{L_j} r_i$ may or may not be less than its reliability expectation ρ_j. If the achieved reliability is strictly less than ρ_j, then we aim to augment its reliability as much as possible to reach the goal ρ_j, subject to the computing resource capacity on each cloudlet in G. Note that such a goal may never be reached due to the lack of demanded computing resource in the MEC network. In the following we show how to calculate the reliability of an admitted request j.

Let R_i be the reliability of network function f_i in SFC_j by placing its (both primary and secondary) VNF instances to cloudlets. The calculation of R_i is as follows. Assuming that there are n_i VNF instances of f_i instantiated in p cloudlets with $1 \leq p \leq n_i$, and let $r_{i,1}, r_{i,2}, \ldots, r_{i,n_i}$ be their reliabilities in these cloudlets, respectively. The accumulative reliability R_i of f_i then is

$$R_i = 1 - \Pi_{l=1}^{n_i}(1 - r_{i,l}). \tag{1}$$

For the sake of convenience, in the rest of discussion we assume that $r_{i,l} = r_i$ for all l with $1 \leq l \leq n_i$, i.e., the reliability of a VNF instance of f_i placed at different cloudlets is identical. This assumption has been widely adopted in literature [7], [8], [11], [13]. Through the placement of both primary and secondary VNF instances of each network function f_i in SFC_j, the reliability u_j of request j finally is

$$u_j = \Pi_{i=1}^{L_j} R_i. \tag{2}$$

To ensure that the reliability expectation ρ_j of request j can be achieved if there are sufficient resources in MEC, we must have

$$\Pi_{i=1}^{L_j} R_i \geq \rho_j. \tag{3}$$

Inequality (3) can be equivalently written as follows.

$$-\sum_{i=1}^{L_j} \log R_i \leq -\log \rho_j. \tag{4}$$

In other words, if the reliability ρ_j is not achievable due to the lack of computing resource at each cloudlet in G, we aim to maximize the value of u_j, or minimize the value of $-\log u_j = -\sum_{i=1}^{L_j} \log R_i$ equivalently.

3.2 Problem Definitions

Assuming that request j has been admitted, and all VNF instances of network functions in its SFC_j have been placed, we term these VNF instances as the *primary VNF instances* of the request. Notice that a primary VNF instance usually is in active status and all its *secondary VNF instances* are in idle statuses. The primary VNF instance communicates with its secondary VNF instances at some pre-defined checking points to replicate itself execution image/status information to its secondary VNF instances, and we assume that such communication delay is negligible. To reduce the response delay of such updatings, all the secondary VNF instances usually are co-placed with its primary VNF instance either in the same cloudlet v or in no more than l-hop cloudlets in $N_l(v)$ from the cloudlet v of its primary VNF instance. We refer to this primary and secondary VNF placement relationships in an MEC network as *the l-hop communication model*, where $l = 1, 2, \ldots, |V| - 1$.

Definition 1. *Given an MEC network $G(V, E)$, each cloudlet $v \in V$ has computing capacity C_v, and the set of network functions $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$, each function $f_i \in \mathcal{F}$ needs $c(f_i)$ computing resource for its implementation in a virtual machine (VM) with $1 \leq i \leq |\mathcal{F}|$. Let r_i be the reliability of f_i in any cloudlet $v \in V$ with $0 < r_i \leq 1$, assume that request j with a service function chain SFC_j has been admitted in G and its reliability expectation is a given value ρ_j. The reliability enhancement for request j is achieved through redundant VNF instance placements to different cloudlets, the service reliability augmentation problem for an admitted request j thus is to maximize its reliability through deploying as many as secondary VNF instances of each VNF instance in SFC_j until its reliability expectation ρ_j is reached, or reaching its best possible reliability, due to running out of computing resources of G.*

Typically, mobile edge clouds provide shared resources for various services. How to allocate limited resources to a set of users, while considering SFC reliability requirement of each request is challenging. We thus study the service reliability augmentation problem for a set Q of admitted requests. Consider that different requests have different SFCs and reliability expectations. If we perform the resource allocation to these admitted requests carelessly, e.g., adopt the utility function in Eq. (2) for a single admitted request, then some admitted requests will easily reach their reliability expectations while others may be far from their reliability expectations due to limited resources in the MEC network. To fairly augment the service reliabilities of admitted requests through fair resource allocation, we adopt the following utility function definition (in Eq. (5)), which aims to achieve the enhanced reliability for each admitted request that is proportional to its reliability expectation.

Denote by u'_j the reliability utility of request q_j in Q, which is defined as follows.

$$u'_j = \frac{\Pi_{i=1}^{L_j} R_{j,i}}{\rho_j} = w_j \cdot \Pi_{i=1}^{L_j} r_{j,i} \quad \text{if } |SFC_j| = L_j, \tag{5}$$

where $R_{j,i}$ is the achieved reliability of function $f_{j,i} \in SFC_j$ of request j through placing its secondary VNF instances to cloudlets, and $w_j = 1/\rho_j$ is the weighting factor of the reliability u'_j of request j with $0 < u'_j \leq 1$, and ρ_j is the given reliability expectation of request q_j.

Definition 2. *We now formally define the service reliability augmentation problem for a set of admitted requests as follows. Given a set Q of admitted requests with each having a different SFC and a reliability expectation in an MEC network $G(V, E)$, the service reliability augmentation problem for a set Q of admitted requests is to augment the reliabilities of all requests in Q by placing secondary VNF instances of each network function in their SFCs to different cloudlets such that the sum $\sum_{j \in Q} u'_j$ of weighted reliabilities of the requests is maximized, subject to the computing capacity on each cloudlet in G, under the l-hop communication model.*

3.3 NP Hardness of the Defined Problem

Theorem 1. *The service reliability augmentation problem for an admitted request with an SFC and a reliability expectation requirements in an MEC network $G = (V, E)$ is NP-hard.*

TABLE 1
Symbols

Symbols	Meaning
$G=(V,E)$	an MEC network G with a set V of access points, and a set E of links
v and C_v	an access point $v \in V$, which may or may not co-located with a cloudlet with computing capacity C_v
$N_l(v)$	l-neighbor set of a cloudlet v
$\mathcal{F}$	the set of all different network functions offered by the system
j and SFC_j	a request j and its service function chain SFC_j
L_j	the length of service function chain SFC_j
ρ_j	the reliability expectation of request j
f_i	a network function within the service function chain SFC_j
r_i	the reliability of a network function $f_i \in SFC_j$
R_i	the reliability of network function $f_i \in SFC_j$ by placing VNF instances to cloudlets
n_i	the number of VNF instances of f_i instantiated in cloudlets
u_j	the reliability of request j
q_j and Q	a request q_j in a set of admitted requests Q
u'_j	the reliability utility of request q_j
C'_v	residual computing capacity of cloudlet v
$c(f_i)$	resource demand of network function f_i
$R_{j,i}$	the achieved reliability of function $f_{j,i} \in SFC_j$ through placing VNF instances to cloudlets
$w_j = 1/\rho_j$	the weighting factor of the reliability u'_j of request j
G_j	a constructed auxiliary directed acyclic graph for primary VNF instance deployment
N_j	the set of cloudlets to host the primary VNF instances of network functions in SFC_j
s_j and t_j	the cloudlets of the source and destination cloudlets of data traffic of request j respectively
A_j	the set of directed edges in G_j from one cloudlet to another in G
$\omega(\cdot,\cdot)$	$\omega : E \mapsto [0,1]$ is a weight function on the edges of G_j
V_l	the set of cloudlets that can instantiate the primary VNF instance of f_l
P and $l(P)$	a shortest path in G_j from s_j to t_j and its length in G_j
G_l	a constructed bipartite graph for heuristic algorithm
M_l	a minimum-cost maximum matching in G_l
C	the cost budget
S	the solution derived from a set of minimum-cost maximum matchings

Proof. See the proof in Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2021.3081681. □

For the sake of convenience, symbols used in this paper are summarized in Table 1.

4 An Admission Framework of a Request With an SFC Requirement

In this section, we provide the admission framework of a single request j with SFC_j and reliability expectation ρ_j, by instantiating the primary VNF instances of each network function in SFC_j to cloudlets in G such that its service reliability is maximized. At this stage, we do not consider instantiating any of secondary VNF instances yet. As the residual computing resources at different cloudlets are different, the primary VNF instance of $f_i \in SFC_j$ can be accommodated by a cloudlet v if its residual computing capacity C'_v is no less than the resource demand of f_i, i.e., $C'_v \geq c(f_i)$.

In this framework, we aim to place the primary VNF instances of network functions in SFC_j such that the service reliability of the request is maximized. Such an admission provides a basic reliability for the request. In the following for the initial VNF instance placement of a request to be admitted, we adopt the similar assumption as we did in [20]. That is, a VNF instance of a request can be placed to a cloudlet if the cloudlet has sufficient computing resource to accommodate all VNF instances in its service function chain SFC_j. Otherwise, the request is not admissible, and should be rejected. To this end, we construct an auxiliary directed acyclic graph (DAG) $G_j = (N_j \cup \{s_j, t_j\}, A_j; \omega)$, where N_j is the set of cloudlets to host the primary VNF instances of network functions in SFC_j, s_j and t_j are the cloudlets of the source and destination cloudlets of data traffic of request j respectively, and A_j is the set of directed edges in G_j from one cloudlet to another in the MEC network G. Function $\omega : E \mapsto [0,1]$ is a weight function on the edges of G_j.

Having constructed G_j, a shortest path in it will correspond a placement scheduling of the primary VNF instances of SFC_j for request j with the maximum reliability. The detailed construction of G_j is given as follows.

There may have multiple candidate cloudlets that can host the primary VNF instance of network function f_i in the service function chain SFC_j if the residual computing capacity of each of them is no less than $c(f_i)$. For the sake of convenience, let V_l be the set of cloudlets that can instantiate the primary VNF instance of f_l with $1 \le l \le L_j$, assuming that network functions in SFC_j are listed as $f_1, f_2, \ldots, f_{L_j}$ with reliability $r_1, r_2, \ldots, r_{L_j}$, respectively. The node set $N_j \cup \{s_j, t_j\}$ of G_j consists of all cloudlets and the source cloudlet s_j, and the destination cloudlet t_j of request j, and $N_j = \cup_{l=1}^{L_j} V_l$.

Algorithm 1. Maximizing Request Reliability by Placing the Primary VNF Instances in its Service Function Chain

Input: An MEC network $G = (V, E)$ with residual computing capacity C'_v at each cloudlet $v \in V$, and a request j with a SFC_j and reliability expectation ρ_j requirements.

Output: Admit request j by placing the VNF instances of SFC_j to cloudlets of G if there are sufficient computing resources in the cloudlets such that the reliability achieved is maximized.

1: Construct a directed auxiliary graph $G_j = (N_j \cup \{s_j, t_j\}, A_j; \omega)$;
2: Find a shortest path P in G_j for s_j to t_j;
3: **if** P exists **then**
4: Place the primary VNF instance of each function in SFC_j to its cloudlet;
5: **return** the solution P;
6: **else**
7: Reject request j;
8: EXIT;
9: **end if**

To ensure that each network functions of f_l is traversed in its specified order in SFC_j, we connect the nodes in $N_j \cup \{s_j, t_j\}$ according to the specified order of their corresponding network functions. That is, we first add a directed edge from s_j to a node $v \in V_1$ with a non-negative weight related to the reliability of running an VNF instance of f_1 in cloudlet v, if the residual computing capacity of v is no less than $c(f_1)$, i.e., $\omega(s_j, v) = -\log r_1$. We then add a directed edge from a node $v \in V_{L_j}$ to t_j and assign its weight 0 if the residual computing capacity of v is no less than $c(f_{L_j})$, i.e., $\omega(v, t_j) = -\log 1 = 0$. We also add a directed edge from a node $u \in V_l$ to a node $v \in V_{l+1}$ and assign its weight to be the negative of the logarithm of the reliability of the VNF instance of f_{l+1} in cloudlet v if the residual computing resources at u and v are no less than $c(f_l)$ and $c(f_{l+1})$, respectively, i.e., $\omega(u, v) = -\log\ r_{l+1}$ with $1 \le l \le L_j - 1$. Thus, the edge set of G_j is $A_j = \{\langle s_j, v\rangle \mid v \in V_1\} \cup \{\langle v, t_j\rangle \mid v \in V_{L_j}\} \cup_{l=1}^{L_j-1} \{\langle u, v\rangle \mid u \in V_l \ \&\ v \in V_{l+1}\}$. Fig. 1 is an example of the constructed auxiliary acyclic graph G_j.

The algorithm for the admission of request j is given in `Algorithm` 1 if there is sufficient resource in G to meet the resource demands of the request.

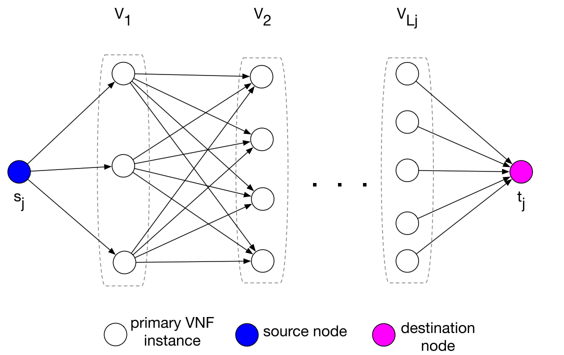


Fig. 1. A constructed auxiliary graph $G_j = (N_j \cup \{s_j, t_j\}, A_j)$, where $V_1, \ldots, V_{L_j}$ represent the sets of candidate nodes for each network function in the service function chain SFC_j of request j with $|SFC_j| = L_j$.

Theorem 2. *Given a request j with SFC_j in G, let $G_j = (N_j \cup \{s_j, t_j\}, A_j; \omega)$ be the auxiliary graph constructed for the admission of request j. If there is not any directed path in G_j from s_j to t_j, then request j is not admissible due to lack of computing resource to accommodate the VNF instances of its SFC_j. Otherwise, a shortest path in G_j from s_j to t_j in terms of the defined edge weight function $\omega(\cdot, \cdot)$ corresponds a feasible VNF instance placement of network functions in SFC_j for request j, and the reliability achieved of this placement is the maximum one. The algorithm takes $O(|V|^2 \cdot L_j)$ time, where $L_j = |SFC_j|$.*

Proof. See the proof in Appendix, available in the online supplemental material. □

5 Integer Linear Program for the Service Reliability Augmentation Problem

In this section, we consider the service reliability augmentation problem for an admitted request j, where all secondary VNF instances of each primary VNF instance in SFC_j must only be placed into the cloudlets no more than l-hops from the cloudlet of the primary VNF instance. For the sake of convenience, in the following we focus on one-hop communication model (i.e., $l = 1$) only, the rest (with $l > 1$) is almost identical with $l = 1$, and omitted.

5.1 Reliability Augmentation of an Admitted Request

Assume that the primary VNF instance of f_i in SFC_j is placed in cloudlet $v \in V$. Let $N_1(v) = \{u_1, u_2, \ldots, u_{d_v}\}$, where d_v is the number of one-hop neighbor cloudlets of cloudlet v in $G(V, E)$ with residual computing capacities $C'_{u_1}, C'_{u_2}, \ldots, C'_{u_{d_v}}$, respectively. Assuming that cloudlet v is cloudlet u_0, i.e., $C'_v = C'_{u_0}$. Let $k_{i,l} = \lfloor \frac{C'_{u_l}}{c(f_i)} \rfloor$ with $0 \le l \le d_v$. For network function f_i in SFC_j, there are at most $d_v + 1$ bins with bin u_l having the residual computing capacity C'_{u_l} and $0 \le l \le d_v$, there are at most $K_i = \sum_{l=0}^{d_v} k_{i,l}$ items of type f_i with each representing one potential secondary VNF instances of f_i.

For each item k_i of type f_i with $1 \le k_i \le K_i$, assume that its primary VNF instance is placed in cloudlet v, then the cost of item k_i placed at any cloudlet $u \in N_l^+(v)$ is $c(f_i, k_i, u) = -\log(R(f_i, k_i) - R(f_i, k_i - 1))$; otherwise, the cost $c(f_i, k_i, u)$ of item k_i placed to cloudlet u is $c(f_i, k_i, u) = M$ for any $u \notin N_l^+(v)$, where M is a sufficiently large positive number, e.g., $M = 100 * \max\{c(f_i, k_i, u) \mid u \in N_l(v) \cup \{v\},\ 0 \le k_i \le K_i,\ \&\ 1 \le i \le L_j\}$, the amount of the computing resource consumed by item k_i is $c(f_i)$. Since there are L_j

different primary VNF instances for SFC_j, there are L_j different types of items.

We then reduce the service reliability augmentation problem for an admitted request j with SFC_j and reliability expectation ρ_j to *a budgeted minimum cost generalized assignment problem* (BMCGAP) that is defined as follows.

Given m bins with bin capacity B_j of bin j, and a set $\mathcal{I}$ of n items, each item $I_i \in \mathcal{I}$ has a positive cost c_{ij} with size s_{ij} if item I_i is packed to bin j, assume that the total cost budget C is given, the problem is to pack as many items in $\mathcal{I}$ as possible to the m bins such that the total cost of packed items is minimized but the total cost is upper bounded by C, subject to the capacity on each bin.

5.2 Overview of the Proposed Algorithm

The reduction proceeds as follows. There are $|V|$ bins with each $v \in V$ having the residual computing capacity of C'_v if $C'_v \neq 0$. Let K_i $(= \sum_{u \in N_l^+(v)} \lfloor \frac{C'_u}{c(f_i)} \rfloor)$ be the maximum number of secondary VNF instances of f_i that can be placed in one or multiple cloudlets u in $N_l^+(v)$ if the cloudlets have sufficient computing resource to accommodate the VNF instances, assuming that the primary VNF instance of f_i is in cloudlet v. Denote by $N_{f,v}$ the set of different types of primary VNF instances of SFC_j placed in cloudlet v.

For each network function f_i in SFC_j, there are K_i items of type f_i with the same computing resource demand $c(f_i)$, they can be placed to at most $d_v + 1$ bins under the assumption of the secondary VNF instance placement, i.e., the bins in $N_l^+(v)$ if $f_i \in N_{f,v}$. However, different items of type f_i will incur different costs, i.e., item k_i of type f_i will incur a cost $c(f_i, k_i, u)$ defined in Eq. (6) if it is placed to bin $u \in N_l^+(v)$; otherwise, it will incur a cost $c(f_i, k_i, u) = M$, where M is defined as a large positive value and $0 \leq k_i \leq K_i$ and $1 \leq i \leq L_j$. There are L_j different types of items.

$$c(f_i, k, u) = -\log\left(R(f_i, k) - R(f_i, k-1)\right) \tag{6}$$

$$\begin{aligned} 1 \leq k \leq K_i,\ u \in N_l^+(v), \text{and } f_i \in N_{f,v}, \\ c(f_i, 0, v) = -\log R(f_i, 0), \quad \text{if } f_i \in N_{f,v} \text{ and } v \in V, \end{aligned} \tag{7}$$

where $R(f_i, 0) = r_i$, $R(f_i, 1) = 1-(1-r_i)(1-r_i) = 1-(1-r_i)^2$, and $R(f_i, k) = 1-(1-r_i)^{k+1}$.

The BMCGAP thus is to pack as many items in $\mathcal{I}$ as possible to the $|V|$ bins to minimize the total cost, subject to the cost budget C $(= -\log \rho_j)$ and the residual computing capacity C'_v on each cloudlet $v \in V$.

5.3 Integer Linear Program Formulation

In the following, we propose an integer linear program (ILP) solution to the service reliability augmentation problem for an admitted request j.

By Inequality (4), the optimization objective is to

$$\text{minimize} \quad \sum_{i=1}^{L_j} -\log R_i \tag{8}$$

subject to the following constraints.

$$\sum_{i=1}^{L_j} -\log R_i \leq -\log \rho_j, \tag{9}$$

$$-\log R_i = \sum_{k_i=0}^{K_i} c(f_i, k_i, u) \cdot x_{i,k_i,u}, \tag{10}$$

$$f_i \in N_{f,v} \text{ and } u \in N_l^+(v), \forall i, 1 \leq i \leq L_j,$$

$$\sum_{u \in N_l^+(v)} x(i, k_i, u) \leq 1, \quad f_i \in N_{f,v} \text{ and } 1 \leq i \leq L_j \tag{11}$$

$$\tag{12}$$

$$\sum_{i=1}^{L_j} \sum_{k_i=0}^{K_i} c(f_i) \cdot x_{i,k_i,u} \leq C'_u, \quad \text{for each } u \in V, \text{and } f_i \in N_{f,v} \tag{13}$$

$$K_i = \sum_{u \in N_l^+(v)} \left\lfloor \frac{C'_u}{c(f_i)} \right\rfloor, \tag{14}$$

$$\text{for each } i \text{ with } 1 \leq i \leq L_j, \text{and } f_i \in N_{f,v},$$

$$x_{i,k_i,u} = \{0, 1\}, \tag{15}$$

$$x_{i,k_i,u} = 0, \quad \text{if } C'_u < c(f_i), \tag{16}$$

$$x_{i,k_i,u} = 0, \quad \text{if } u \in V \setminus N_l^+(v) \,\&\, f_i \in N_{f,v}, \tag{17}$$

$$x_{i,k_i,u} = 0, \quad \text{if } c(f_i, k_i, u) = M, \tag{18}$$

where R_i is the achieved reliability of network function $f_i \in SFC_j$ through placing multiple VNF instances to different cloudlets, which is defined in Eq. (1). Note that when R_i becomes larger through placing more its VNF instances into the network, the value of $-\log R_i$ (> 0) becomes smaller and $0 < R_i \leq 1$. Variable $x_{i,k_i,u}$ is a binary variable. If it is 1, then, the k_ith secondary VNF instance of f_i is placed to cloudlet $u \in V$.

Constraint (9) ensures that the final reliability of request j is no greater than its expectation $-\log \rho_j$. Constraint (10) ensures that the number K_i of secondary VNF of f_i is as large as possible, thus, the value of $-\log R_i$ becomes smaller. Constraint (11) ensures that each item can be placed to no more than one cloudlet. Constraint (13) ensures that different secondary VNF instances at each cloudlet u is no more than its capacity. Constraint (14) calculates the maximum number of possible secondary VNF instance for each $f_i \in N_{f,v}$. Constraint (16) ensures that none of any secondary VNF instance is placed to a cloudlet without its demanded computing resource. Constraint (17) ensures that any secondary VNF instance of a primary VNF instance placed in cloudlet v will not be placed to a cloudlet with more than l-hops from the cloudlet of its primary VNF instance. Constraint (18) is equivalent to Constraint (16), which implies that the VNF instance corresponding item k_i of type f_i cannot be placed into cloudlet u.

5.4 Algorithm Analysis

We first show the property of the cost function $c(\cdot,\cdot,\cdot)$ defined in Eq. (6) by Lemma 1. We then analyze the property of the exact solution of the ILP.

Lemma 1. *For the defined cost function in Eq. (6), we have*

$$(i)\quad c(f_i,k,u) > 0, \qquad \text{for any } k \geq 0 \text{ and } f_i \in N_{f,u}, \tag{19}$$

$$(ii)\quad c(f_i,k',*) > c(f_i,k,*), \quad \text{if } k' > k \geq 1, \\ f_i \in N_{f,v}, \text{and } * \text{ is any cloudlet in } N_l^+(v). \tag{20}$$

Proof. (i) When $k=0$, $c(f_i,0,u)=r_i>0$. When $k\geq 1$, $R(f_i,k)=1-(1-r_i)^{k+1}$ and $R(f_i,k-1)=1-(1-r_i)^k$, we then have $R(f_i,k)>R(f_i,k-1)$ due to $0<r_i<1$, and $c(f_i,k,u)=-\log(R(f_i,k)-R(f_i,k-1))>0$.

(ii) We show that $c(f_i,k',*)>c(f_i,k'-1,*)$ as follows.

$$\begin{aligned} & c(f_i,k',*)-c(f_i,k'-1,*) \\ &= -\log(R(f_i,k')-R(f_i,k'-1)) \\ &\qquad -(-\log(R(f_i,k'-1)-R(f_i,k'-2))) \\ &= \log(R(f_i,k'-1)-R(f_i,k'-2)) \\ &\qquad -(\log(R(f_i,k')-R(f_i,k'-1))) \\ &= \log\frac{1}{(1-r_i)} \\ &> 0, \quad \text{since } \frac{1}{1-r_i} > 1. \end{aligned} \tag{21}$$

By Inequality (21), we have

$$c(f_i,k',*) > c(f_i,k'-1,*) > \ldots > c(f_i,k,*), \\ \text{if } k' > k. \tag{22}$$

The lemma then follows. □

Lemma 2. *Given an exact solution delivered by the ILP, we claim that if $x_{i,k_i,*}=1$ with k_i is the largest value in the solution that is no greater than K_i, then $x_{i,k'_i,*}=1$ for any $k'_i \leq k_i$, where $*$ represents any cloudlet $u \in N_l^+(v)$ and $f_i \in N_{f,v}$.*

Proof. We show the claim by contradiction. Assume that there exists $x_{i,k_i,u}=1$ while $x_{i,k'_i,u'}=0$ with $k'_i<k_i$ in the solution with $u' \in N_l^+(v)$. Following the cost definition and Lemma 1, $c(f_i,k_i,u)>c(f_i,k'_i,u')$ but both items k'_i and k_i have the same size $c(f_i)$. Another better solution with a smaller cost can be obtained, by replacing item k_i with item k'_i. This contradicts that the solution obtained by the ILP is the optimal one with the minimum cost. The lemma then follows. □

6 Randomized Algorithm for the Service Reliability Augmentation Problem

In this section, we devise a randomized algorithm for the service reliability augmentation problem based on the ILP formulation. Following the random rounding technique [25], we first relax the ILP to a Linear Program (LP). An optimal solution of the LP can be obtained in polynomial time. We then round the fractional solution of the LP with probability to a 0/1 integer solution. We finally show that the 0/1 integer solution is very likely to be a feasible solution of the service reliability augmentation problem with high probability.

The detailed randomized algorithm for the service reliability augmentation problem of an admitted request j is given in `Algorithm` 2.

Algorithm 2. A Randomized Algorithm for the Service Reliability Augmentation Problem of an Admitted Request j With the Assumption That all the Secondary VNF Instances of a Primary VNF Instance in Cloudlet $v \in V$ Can Only be Placed into the Cloudlets in $N_l^+(v)$ for a Fixed Integer l With $1 \leq l \leq |V|-1$

Input: An MEC network $G=(V,E)$, and request j with SFC_j and reliability expectation ρ_j, assuming that the primary VNF instances of SFC_j have been placed into the cloudlets in G.

Output: Find a solution for the problem, where all the secondary VNF instances of each primary VNF instance will be placed to the cloudlets no more than l-hops from the cloudlets of their primary VNF instances to maximize the reliability of request j until either reaching its reliability expectation ρ_j or as large as possible.

Admit request j, by placing the primary VNF instance of each function in SFC_j through invoking `Algorithm` 1;
if $\Pi_{l=1}^{L_j} r_l \geq \rho_j$ **then**
  meeting the reliability expectation of request j, EXIT;
end if ;
Solve the relaxed version LP of ILP (8) in polynomial time;
Let $\widetilde{OPT}$ be the optimal solution of the LP and $\tilde{x}_{i,k_i,u}$ the value of each variable $x_{i,k_i,u}$, where $\tilde{x}_{i,k_i,u} \in [0,1]$;
An integer solution $\hat{x}_{i,k_i,u}$ can be obtained by the randomized rounding approach in [25]. That is, $\hat{x}_{i,k_i,u}$ is set to 1 with probability of $\tilde{x}_{i,k_i,u}$; otherwise, $\hat{x}_{i,k_i,u}$ is set to 0; The choice is performed in an exclusive manner, with Constraint (11): for each $u, \forall u \in N_l^+(v)$, exactly one of the variables $\hat{x}_{i,k_i,u}$ is set to one 1, and the rest are set to 0s. This random choice is made independently for all u;
A candidate integer solution $\hat{S}$ can be derived based on $\hat{x}_{i,k_i,u}$, which will be a feasible solution to the ILP with high probability.

6.1 Algorithm Analysis

The rest is to analyze the approximation ratio of `Algorithm 2` and the computing resource violation on each cloudlet. We start with the following lemma.

Lemma 3. *(Chernoff bounds) Given n independent variables $x_1,x_2,\ldots,x_n$ where $x_i \in [0,1]$, let $\mu=\mathbb{E}[\sum_{i=1}^n x_i]$. Then,*

(i) Upper Tail: $Pr[\sum_{i=1}^n x_i \geq (1+\beta)\mu] \leq e^{\frac{-\beta^2\mu}{2+\beta}}$ *for all* $\beta>0$,

(ii) Lower Tail: $Pr[\sum_{i=1}^n x_i \leq (1-\beta)\mu] \leq e^{\frac{-\beta^2\mu}{2}}$ *for all* $0<\beta<1$.

We then have the following theorem.

Theorem 3. *Given an MEC network $G(V,E)$ and a request j with SFC_j and reliability expectation ρ_j, there is a randomized* *algorithm,* `Algorithm 2`*, with high probability of* $\min\{1-\frac{1}{N}, 1-\frac{1}{|V|^2}\}$ *for the service reliability augmentation problem. The expected approximation ratio of the algorithm is* $(1/P^*)^{1-\frac{2}{\Lambda}}$*, and the computing resource violation ratio at any cloudlet is no more than twice its capacity, provided that* $P^* \geq \frac{1}{N^{3\Lambda/\log e}}$ *and* $min_{v\in V}\{C_v\} \geq 6\Lambda \ln V$*, where* $N = \sum_{i=1}^{L_j} K_i \leq \lceil \frac{L_j \cdot C_{max} \cdot d_{max}}{c_{min}} \rceil$*,* K_i *is the maximum number of secondary VNF instances for function* $f_i \in SFC_j$*,* Λ *is a constant strictly greater than 2,* P^* *is the optimal reliability of request* j *in* G*, and* Λ *is a constant defined in Eq. (29).*

Proof. See the proof in Appendix, available in the online supplemental material. □

7 Heuristic Algorithm for the Service Reliability Augmentation Problem

In this section, we propose an efficient heuristic algorithm for the problem that delivers a feasible solution without any computing capacity violations.

7.1 Overview of the Algorithm

The basic idea is to augment the reliability of request j through constructing a series of bipartite graphs $G_0, G_1, \ldots, G_l$. For each bipartite graph G_l, find a minimum-cost maximum matching M_l from G_l, which corresponds to a subset of secondary VNF instance placement to their matched cloudlets without violating the computing capacity of any cloudlet. This procedure continues until either the total cost reaches the cost budget C, or no more computing resource is available for further secondary VNF instance placement.

7.2 Algorithm

Following the problem optimization objective, we aim to augment the reliability of request j by placing as many secondary VNF instances as possible to cloudlets while minimizing the placement costs, subject to the cost budget C and computing resource capacity on each cloudlet.

We construct a series of auxiliary bipartite graphs. We start with graph $G_0 = (V, \mathcal{I}, E_0; c)$ as follows. Each node $v \in V$ has a residual computing capacity C'_v, and $\mathcal{I}$ is the set of all possible secondary VNF instances of VNFs in SFC_j, i.e., $\mathcal{I} = \cup_{i=1}^{L_j} \cup_{k_i=0}^{K_i} \{I_{k_i}\}$, there is an edge $(u, I_{k_i}) \in E_0$ in G_0 between nodes $u \in V$ and $I_{k_i} \in \mathcal{I}$ with cost $c(f_i, k_i, u)$ if $f_i \in N_{f,v}$, $u \in N_l^+(v)$, and $C'_u \geq c(f_i)$. We then find a minimum cost maximum matching in the auxiliary graph. This procedure continues until no matching exists in the auxiliary graph. The detailed algorithm is presented in `Algorithm 3`.

7.3 Algorithm Analysis

In the following, we first show that the solution delivered by `Algorithm 3` is feasible. We then analyze the time complexity of the proposed algorithm.

Lemma 4. *For any function* f_i *in* SFC_j *of request* j*, assume that its primary VNF instance is placed at cloudlet* v*, if there are* K'_i *items of this type function that have been packed into cloudlets in* $N_l^+(v)$ *with* $0 \leq K'_i \leq K_i$*, by* `Algorithm 3`*, then, these packed* K'_i *items must be the top-*K'_i *smallest items in terms of the defined cost.*

Proof. Assume that there is an item k_i for f_i which is placed in a bin $u \in N_l^+(v)$ that is not one of the first K'_i smallest items of this type. Let k'_i be one of the top-K'_i smallest items, i.e., $k'_i \leq K'_i$ while $k_i > K'_i$. We replace item k_i by item k'_i into bin u of item k_i, there does not incur any change in terms of the amounts of computing resource consumption for either of them. However, the amount of cost reduced by this replacement is $c(f_i, k_i, v) - c(f_i, k'_i, v) > 0$ by Lemma 1, as $k_i > k'_i$. The lemma then follows. □

Theorem 4. *Given an MEC network* $G(V, E)$ *and an admitted request* j *with* SFC_j *and reliability expectation* ρ_j*, each cloudlet* $v \in V$ *has residual computing capacity* C'_v*. There is an efficient algorithm,* `Algorithm 3`*, for the service reliability augmentation problem of an admitted request* j*, under the assumption that all the secondary VNF instances of each primary VNF instance must be placed into the cloudlets no more than l-hops from the cloudlet of the primary VNF instance, where* l *is a fixed integer with* $1 \leq l \leq |V| - 1$*. The the time complexity of* `Algorithm 3` *is* $O((N^3 + |V|^3) \cdot \log_{\frac{d_{min}}{d_{min}+1}} N)$*, where* $N = \lceil \sum_{i=1}^{L_j} K_i \rceil \leq \lceil \frac{L_j \cdot C_{max} \cdot d_{max}}{c_{min}} \rceil$*,* $d_{min} = \min\{d_v \mid v \in V\}$*,* $d_{max} = \max\{d_v \mid v \in V\}$*,* $C_{max} = \max_{v\in V}\{C_v\}$*,* $c_{min} = \min\{c(f_i) \mid f_i \in SFC_j\}$*, and* $L_j = |SFC_j|$*.*

Proof. See the proof in Appendix, available in the online supplemental material. □

8 Algorithm for the Service Reliability Augmentation Problem for a Set of Admitted Requests

In this section, we show how to solve the service reliability augmentation problem for a set Q of admitted requests through invoking the proposed algorithm `Algorithm 3`, assuming that $Q = \{q_1, q_2, \ldots, q_{|Q|}\}$ and each request q_j has a SFC_j and a reliability expectation ρ_j with $1 \leq j \leq |Q|$.

8.1 Algorithm

For a given request $q_j \in Q$, assume that $f_{j,1}, f_{j,2}, \ldots, f_{j,L_j}$ is the sequence of the network functions in SFC_j with $|SFC_j| = L_j$, and further assume that the primary VNF instance of $f_{j,i}$ has been placed in cloudlet $v_{j,i} \in V$, where we use $v_{j,i}$ to represent cloudlet $v_p \in V$ in which the primary VNF instance of $f_{j,i}$ is placed, where $1 \leq i \leq L_j$ and $1 \leq j \leq |Q|$. For each request $q_j \in Q$ with SFC_j, if the primary VNF instance of $f_{j,i}$ is placed in cloudlet v, then there are at most $K_{j,i}$ items of type $f_{j,i}$ that can be placed into the cloudlets in $N_l(v)$, where $K_{j,i} = \sum_{u\in N_l(v)\cup\{v\}} \lfloor \frac{C'_u}{c(f_{j,i})} \rfloor$, i.e., for each item $k_{j,i}$ of type $f_{j,i}$, if it is packed into cloudlet $u \in N_l^+(v) \subseteq V$, then the cost incurred by this placement $c(f_{j,i}, k_{j,i}, u)$ is defined as follows.

$$c(f_{j,i}, k_{j,i}, u) = -\log(R'(f_{j,i}, k_{j,i}) - R'(f_{j,i}, k_{j,i} - 1)), \quad \text{if } u \in N_l^+(v) \tag{23}$$

$$c(f_{j,i}, k_{j,i}, u) = M, \quad \text{otherwise } (u \in V \backslash N_l^+(v)), \tag{24}$$

and

$$R'(f_{j,i},k) = 1-(1-r_{j,i})^k, \quad \forall k \quad \text{with } 1 \le k \le K_{j,i}, \tag{25}$$

where $R'(f_{j,i},0) = r_{j,i}$, $R'(f_{j,i},1) = 1-(1-r_{j,i})(1-r_{j,i}) = 1-(1-r_{j,i})^2$, and $R'(f_{j,i},k) = 1-(1-r_{j,i})^{k+1}$, $0 \le k_{j,l} \le K_{j,i}$ for all i and j with $1 \le i \le L_j$ and $1 \le j \le |Q|$.

The problem then is to

$$\text{maximize} \qquad \sum_{j=1}^{|Q|} u'_j, \tag{26}$$

where u'_j is defined in Eq. (5). The optimization objective (26) is equivalent to minimize

$$\begin{aligned} &-\sum_{j=1}^{|Q|} \log u'_j \\ &= -\sum_{j=1}^{|Q|} \left(\sum_{i=0}^{L_j} (\log R'_{j,i} - \log \rho_j) \right) \\ &= \sum_{j=1}^{|Q|} \left[\log \rho_j - \sum_{i=0}^{L_j} \sum_{k_{j,i}=0}^{K_{j,i}} c(f_{j,i},k_{j,i},u) x_{j,k_i,u} \right]^+, \\ &\qquad u \in N_l(v) \cup \{v\} \text{ and } f_{j,i} \in N_{f,v}, \end{aligned} \tag{27}$$

where $[a]^+ = 0$ if $a \le 0$; otherwise, $[a]^+ = a$.

The algorithm for the service reliability augmentation problem for a set Q of admitted requests is almost identical to Algorithm 3. That is, we reduce the problem to a minimum cost generalized assignment problem (GAP) [22]. The only difference is that the total cost budget is not given. The detailed algorithm and its complexity analysis thus are omitted. We refer to this algorithm as Algorithm 4.

There are $|V|$ bins, and each bin $v \in V$ has residual computing capacity C'_v. For each request $q_j \in Q$, there are $K_j = \sum_{i=1}^{L_j} K_{j,i}$ items in total. The minimum cost GAP thus is to pack as many as $\sum_{j=1}^{|Q|} K_j$ items to the $|V|$ bins such that the total cost is minimized, subject to the residual computing capacity on each bin $v \in V$, i.e., the problem is to

$$\text{minimize} \qquad -\sum_{j=1}^{|Q|} \sum_{k_i=0}^{L_j} w_j \cdot \log R_{j,i} \le -\log |Q|, \tag{28}$$

Since $\frac{\Pi_{i=1}^{L_j} r_{j,i}}{\rho_j} \le 1$, $\log R - \log \rho_j < 0$ with $R = \Pi_{i=1}^{L_j} r_{j,i}$.

Theorem 5. *Given an MEC network $G(V,E)$ and a group Q of admitted requests with each request $q_j \in Q$ with a SFC_j and a reliability expectation ρ_j, each cloudlet $v \in V$ has computing capacity C_v, there is an algorithm,* Algorithm 4, *for the service reliability augmentation problem for a set Q of admitted requests, under the assumption that all secondary VNF instances of each primary VNF instance must be placed in the cloudlets in $N_l(v)$ if the primary VNF instance is placed in cloudlet $v \in V$ with a fixed integer l and $1 \le l \le |V|-1$.* Algorithm 4 *takes $O(|Q| \cdot (N^3 + |V|^3) \cdot \log_{\frac{d_{min}}{d_{min}+1}} N)$, where $N = \max_{1 \le j \le |Q|}\{\lceil \sum_{i=1}^{L_j} K_{j,i} \rceil \le \lceil \frac{L_j \cdot C_{max} \cdot d_{max}}{c_{min}} \rceil\}$, $d_{min} = \min\{d_v \mid v \in V\}$, $d_{max} = \max\{d_v \mid v \in V\}$, $L_{max} = \max\{L_j \mid q_j \in Q, \& \ L_j = |SFC_j|\}$, $C_{max} = \max\{C_v \mid v \in V\}$, $c_{min} = \min\{c(f_i) \mid f_i \in \mathcal{F}\}$, $d_{max} = \max_{v \in V}\{|N_l(v)| \mid v \in V\}$.*

Algorithm 3. Heuristic Algorithm for the Service Reliability Augmentation Problem Under the Assumption That all Secondary VNF Instances of a Primary VNF Instance Must be Placed into the Cloudlets no More Than l-hops From the Cloudlet of the Primary VNF Instance

---

Input: An MEC network $G(V,E)$ with residual computing capacity C'_v and an admitted request j with the primary VNF instances of its SFC_j placed and reliability expectation ρ_j.

Output: Augment the reliability of request j by placing all the secondary VNF instances of each primary VNF instance to the cloudlets no more than l-hops from the primary VNF instance, subject to the residual computing capacity on each cloudlet $v \in V$ and the total placement budget $C = -\log \rho_j$.

Place the primary VNF instance of each function in SFC_j of request j, by invoking Algorithm 1;
if $\Pi_{l=1}^{L_j} r_i \ge \rho_j$ **then**
  The admission of request j meets its reliability expectation ρ_j; EXIT;
end if ;
Construct the initial bipartite graph $G_0(V,\mathcal{I},E_0;c)$;
$S \leftarrow \emptyset$; /* the solution */
$l \leftarrow 1$; $G_1 \leftarrow G_0$; $E_1 \leftarrow E_0$;
while ($c(S) < C$ and $E_l \ne \emptyset$) **do**
  Find a minimum-cost maximum matching M_l in G_l, by the Hungarian algorithm;
  $S \leftarrow S \cup M_l$;
  $C'_v \leftarrow C'_v - c(f_i)$ if $\exists (v, I_{k_i}) \in M_l$ for each $v \in V$;
  $l \leftarrow l+1$; $\mathcal{I} \leftarrow \mathcal{I} \setminus \{I_{k_i} \mid (v,I_{k_i}) \in M_l\}$;
  Construct the next bipartite graph $G_l = (V', \mathcal{I}, E_l; c)$, where $V' = \{v \mid v \in V \text{ and } C'_v \ne 0\}$; E_l is the set of edges between the nodes in V' and $\mathcal{I}$, and an edge $(v, I_{k_i}) \in E_l$ if $f_i \in N_{f,v}$ and $C'_v \ge c(f_i)$;
  $c(S) \leftarrow \sum_{(v,I_{k_i}) \in S} c(f_i,k_i,v)$; /* the total cost of the solution */
end while
return Solution S.

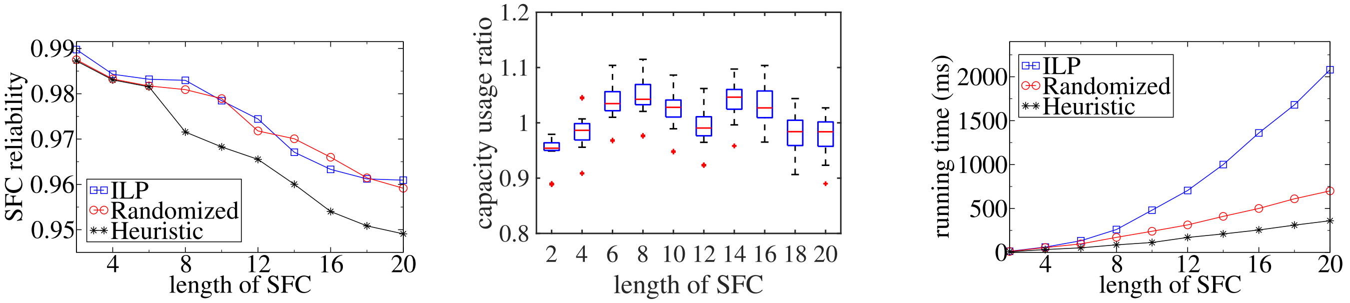
Proof. The proof body is almost identical to the one in the proof of Theorem 4, omitted. □

9 Performance Evaluation

In this section, we evaluate the performance of the proposed algorithms for the service reliability augmentation problem. We also investigate the impact of parameters on the performance of the proposed algorithms.

9.1 Experiment Settings

We consider an MEC network $G = (V,E)$ that consists of 200 APs, in which the number of cloudlets is 5 percent of the network size, and the cloudlets are randomly co-located with some of the APs. Each network topology is generated using the widely adopted approach due to GT-ITM [6]. The computing capacity of each cloudlet ranges from 4,000 to 8,000 MHz [12]. The number $|\mathcal{F}|$ of different types of network functions is set at 30. The computing resource demand of each network function is set from $200\ MHz$ to $400\ MHz$ [2]. For each



(a) The reliability of a request

(b) The capacity usage ratio of cloudlets by algorithm `Randomized`

(c) The running times of different algorithms

Fig. 2. Performance of algorithms `ILP`, `Randomized`, and `Heuristic`, by varying the SFC length of a request from 2 to 20.

generated request j, the length $|SFC_j|$ of its service function chain SFC_j is set between 3 and 10, and each network function is randomly drawn from the $|\mathcal{F}|$ types. Each VNF instance in the primary SFC deployed randomly into cloudlets. We assume that its secondary VNF instances can be placed in cloudlets no more than one hop from the primary VNF instance, i.e., $l = 1$. The running time of an algorithm is obtained on a machine with 3.4 GHz Intel i7 Quad-core CPU and 16 GB RAM. Unless otherwise specified, these parameters will be adopted in the default setting.

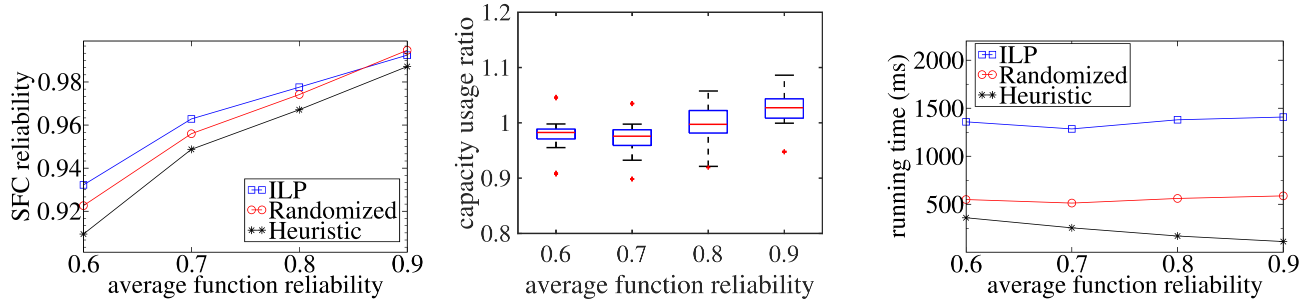
9.2 Performance Evaluation of Algorithms for Service Provisioning of an Admitted Request

In the following, we evaluate the proposed algorithms, ILP, `Algorithms` 2 and 3. For simplicity, we refer to `Algorithms` 2 and 3 as `Randomized`, and `Heuristic`, respectively. For each request with a given length of SFC, 1,000 requests with the same SFC length of the request are randomly generated for each set of experiments. Each value in figures is the mean of the results of these 1,000 trials.

We first evaluate the performance of algorithms `Randomized` and `Heuristic` against the exact solution delivered by the ILP for the service reliability augmentation problem, by varying the SFC length of a request from 2 to 20, while fixing the residual computing capacity of each cloudlet at 25 percent, and the reliability r_i of each network function f_i in the SFC is randomly drawn between 0.8 and 0.9. Fig. 2 illustrates the achieved service function chain reliability, the running times of the three mentioned algorithms, and the ratio of the cloudlet computing capacity usage for algorithm `Randomized`. It can be seen from Fig. 2a that algorithms `Randomized` and `Heuristic` can achieve a near optimal service function chain reliability, i.e., the reliabilities delivered by algorithms `Randomized` and `Heuristic` are no less than 97.82 and 96.03 percent of the optimal one, respectively. Notice that, the reliability delivered by algorithm `Randomized` in some cases is higher than that by ILP, due to allowing violating resource capacity constraints. This has been demonstrated in Fig. 2b. Fig. 2b depicts the average, the minimum, and the maximum computing capacity usage ratio by algorithm `Randomized`. Fig. 2c plots the running time curves of the three mentioned algorithms. It can be seen that the running times of algorithms `Randomized` and `Heuristic` are much less than that of ILP, while their solutions are almost comparable to the exact one by the ILP. With the increase on the problem size, the running time of the ILP grows rapidly, and the running time gap between the ILP and the other two algorithms becomes larger and larger. It must be mentioned that the running time of algorithm `Heuristic` is the least one among the three comparison algorithms for all cases.

We then study the performance of the three mentioned algorithms, by varying the reliability of each network function from 0.6 to 0.9 while keeping other parameters not been changed. Specifically, the reliability of a network function is drawn from intervals $[0.55\ 0.65)$, $[0.65\ 0.75)$, $[0.75\ 0.85)$, and $[0.85\ 0.95]$, respectively. The results delivered by different algorithm are shown in Fig. 3. It can be seen from Fig. 3a that when the network function reliability of each VNF instance increases, the reliability of the service function chain reliability increases at the same time, and the performance gap between the three algorithms becomes smaller. For example, when the average network function reliability is 0.6, algorithm `Randomized` achieves a service function chain reliability 2.03 percent less than that by the ILP, and when the average network function reliability is 0.8, algorithm `Randomized` achieves a service function chain reliability 0.79 percent less than that by the ILP. Similar performance can be observed for algorithm `Heuristic` as well. Notice that the service function chain reliability achieved by algorithm `Randomized` can be higher than that by the ILP due to possible computing resource violation, which is demonstrated in Fig. 3b. Fig. 3c plots the running time curves of the three algorithms, where algorithm ILP takes the longest running time, and algorithm `Heuristic` takes the least running time.

We finally evaluate the performance of the three mentioned algorithms, by varying the ratio of residual computing capacity of cloudlets to its capacity, while keeping the other parameters unchanged. Fig. 4a illustrates the service function chain reliability achieved by different algorithms. It can be seen that when the network has a relatively abundant computing resource, i.e., when there are 50 or 100 percent of residual computing capacities of each cloudlet, algorithms `Randomized` and `Heuristic` can achieve nearly optimal reliability for each request. However, when the residual computing resource in the network becomes less and less, the service function chain reliability decreases. For example, when the network has 50 percent the residual computing capacity per cloudlet, the three comparison algorithms `ILP`, `Randomized`, and `Heuristic` can deliver solutions with service function chain reliabilities by 98.30, 97.12, and 96.42 percent, respectively; when the computing resource is seriously shortage in network wide, i.e., when there is $1/16$ of the residual computing capacity per cloudlet, the service function chain reliabilities achieved by the three algorithms are 66.07, 62.90, and 60.19 percent,

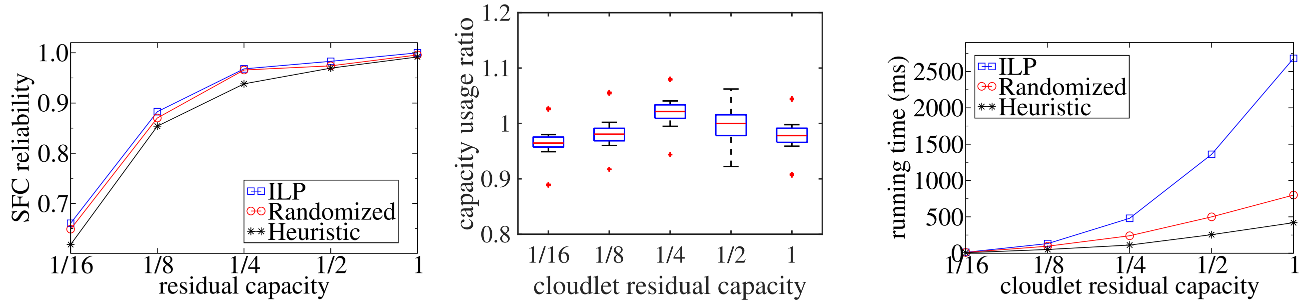


(a) The reliability of a request

(b) The capacity usage ratio of cloudlets by algorithm `Randomized`

(c) The running times of different algorithms

Fig. 3. Performance of algorithms `ILP`, `Randomized`, and `Heuristic`, by varying the network function reliability from 0.6 to 0.9.



(a) The reliability of a request

(b) The capacity usage ratio of cloudlets by algorithm `Randomized`

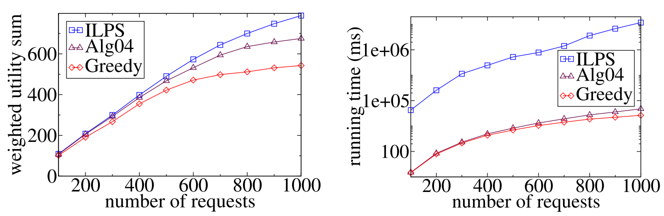
(c) The running times of different algorithms

Fig. 4. Performance of algorithms `ILP`, `Randomized`, and `Heuristic` by varying the residual computing capacity of each cloudlet from $1/16$ to 1.

respectively. The reason behind this is that the amounts of available computing resource in the network determine the number of secondary VNF instances of each primary VNF instance can be instantiated. Fig. 4b depicts the computing capacity usage ratio by algorithm `Randomized`, and Fig. 4c depicts the running time curves of the three algorithms, from which we can see that with the increase on the residual computing capacity, more secondary VNF instances can be instantiated, and the running times of all the three algorithms increase. Similarly, algorithm `ILP` takes the longest running time while algorithm `Heuristic` takes the least running time.

9.3 Performance Evaluation of Algorithms of Service Provisioning for a Set of Admitted Requests

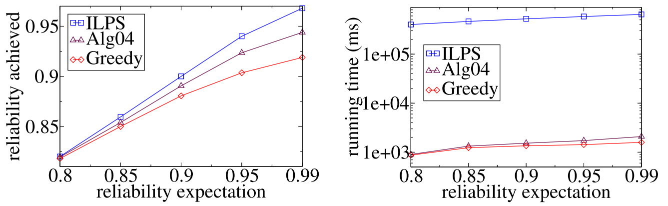
We now investigate the performance of the proposed algorithm `Algorithm` 4 against its benchmark algorithms `ILPS` and `Greedy` for a set of requests, where algorithm `Greedy` examines each request one by one and invokes subroutine `Heuristic` for each request until all requests in the set meet their reliability expectations or no more VNF instances can be instantiated in any cloudlet without violating the computing capacity of the cloudlet. `ILPS` is the integer linear program for the service reliability augmentation problem for a set of requests. We evaluate the three mentioned algorithms by varying the number of requests from 100 to 1,000, and setting the request reliability expectations between 0.9 and 0.99 while fixing the percentage of cloudlets to the network size as 50 percent. The reliability of each network function varies from 0.85 to 0.95. Fig. 5 illustrates the accumulative weighted utility sum of requests and running time curves of the three algorithms. It can be seen from Fig. 5a that all algorithms deliver solutions with increasing accumulative weighted utility, along with the increase on the number of requests. `Algorithm` 4 can achieve at lease 88.74 percent of the optimal solution. For example, when the number of requests is 500, `Algorithm` 4 achieves 96.44 percent of the accumulative weighted utility sum of the optimal solution. It can also be seen from this figure that `Algorithm` 4 outperforms algorithm `Greedy`. Specifically, when the number of requests reaches 600, `Algorithm` 4 delivers 14.86 percent more weighted utility sum than that of algorithm `Greedy` when the number of requests is 600, while their performance gap increases to 25.49 percent when the number of requests is 1,000. The rationale behind is that `Algorithm` 4 strives for the better fairness on reliability augmentation among all requests, while algorithm `Greedy` only examines requests one by one and always tries to instantiate more VNF instances in its nearby cloudlets greedily to maximize the reliability augmentation of each single request. Fig. 5b plots the running time curves of the three algorithms. It can be seen that `ILPS` takes a much longer time to deliver an optimal solution, while `Algorithm` 4 delivers a near optimal solution in a much shorter time. It must also be mentioned that the running time of `ILPS` becomes prohibitive high with the increase of the number of requests, and the solution is no longer achievable



(a) The accumulative weighted utility sum of requests

(b) The running times of different algorithms

Fig. 5. Performance of algorithms `Alg04`, `ILPS`, and `Greedy` by varying the number of requests from 100 to 1,000.



(a) The average request reliabilities achieved

(b) The running times of different algorithms

Fig. 6. Performance of algorithms `Alg04`, `ILPS`, and `Greedy` by varying the average request reliability expectations from 0.8 to 0.99.

when the number of requests reaches 1,000. Although algorithm Greedy has the least running time, its solution results in the least accumulative weighted utility.

We also study the performance of the three mentioned algorithms for the reliability augmentation problem for a set of requests, by varying the average of request reliability expectations between 0.8 and 0.99, while drawing reliability of each network function from [0.75, 0.9] and fixing the number of requests as 500. Fig. 6 depicts the performance curves of the three algorithms. It can be seen from Fig. 6a that `ILPS` achieves the highest average request reliabilities. `Algorithm` 4 outperforms algorithm `Greedy` in all cases, and the performance gap between them becomes larger with the growth of the average request reliability expectation. For example, the average request reliability achieved by `Algorithm` 4 is 4.81 and 7.64 percent more than those by algorithm `Greedy`, when the average request reliability expectation is 0.9 and 0.99, respectively, from which the necessity of considering request fairness can be justified. Fig. 6b depicts the running time curves of the mentioned algorithms, where `ILPS` takes the much more running time than the other two, while algorithm `Greedy` takes less running time than that of `Algorithm` 4, as less potential VNF instances to be deployed are considered in each round of maximum matching and the computing resource in each cloudlet runs out quickly.

10 Conclusion

In this paper, we studied a novel reliability augmentation problem for an admitted request (or a set of admitted requests) with a service function chain and reliability expectation requirements in an MEC network, by enhancing its service reliability through placing redundant VNF instances into different cloudlets. We first showed that the problem is NP-hard, and provided a framework for admissions of such requests. We then proposed an integer linear program solution and a randomized algorithm with a good approximation ratio for the problem, under the assumption that all the secondary VNF instances must be placed into the cloudlets no more than l hops from the cloudlets of their primary VNF instances. We also devised an efficient heuristic algorithm for the problem through reducing the problem to a series of minimum-cost maximum matching problems. Furthermore, we proposed an algorithm for the reliability augmentations of a set of admitted requests by extending the solutions for a single admitted request. We finally evaluated the performance of the proposed algorithms through experimental simulations. Experimental results demonstrate that the proposed algorithms are promising, and their empirical results are superior to their analytical counterparts.

Acknowledgments

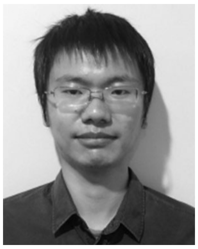
The authors would like to thank the three anonymous referees for their constructive suggestions, which helped us to improve the quality and presentation of the paper greatly. The work of Weifa Liang and Yu Ma was supported by the Australian Research Council Discovery through Project DP200101985. The work of Wenzheng Xu was supported by in part by the National Natural Science Foundation of China (NSFC) under Grant 61602330 and in part by the Sichuan Science and Technology Program under Grants 2018GZDZX0010 and 2018GZDZX0020. The work of Zichuan Xu was supported in part by the National Natural Science Foundation of China under Grant 61802048 and in part by the Xinghai Scholar Program in Dalian University of Technology, China. The work of Xiaohua Jia was supported by the Research Grants Council of Hong Kong through Project CityU 11214316.

References

[1] S. Aidi, M. F. Zhani, and Y. Elkhatib, "On improving service chains survivability through efficient backup provisioning," in *Proc. Int. Conf. Netw. Serv. Manage.*, 2018, pp. 108–115.
[2] Amazon Web Services, Inc., "Amazon EC2 instance configuration," 2018. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-ec2-config.html
[3] D. Chemodanov, P. Calyam, and F. Esposito, "A near optimal reliable composition approach for geo-distributed latency-sensitive service chains," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1792–1800.
[4] W. Ding, H. Yu, and S. Luo, "Enhancing the reliability of services in NFV with the cost-efficient redundancy scheme," in *Proc. Int. Conf. Commun.*, 2017, pp. 1–6.
[5] H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molish, "Approximation algorithms for the NFV service distribution problem," in *Proc. INFOCOM IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.
[6] GT-ITM 2018. [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/
[7] J. Fan, C. Guan, Y. Zhao, and C. Qiao, "Availability-aware mapping of service function chains," in *Proc. INFOCOM IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.
[8] J. Fan, M. Jiang, and C. Qiao, "Carrier-grade availability-aware mapping of service function chains with on-site backups," in *Proc. IEEE/ACM Int. Symp. Qual. Serv.*, 2017, pp. 1–10.
[9] J. Fan *et al.*"A framework for provisioning availability of NFV in data center networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2246–2258, Oct. 2018.
[10] B. Han, V. Gopalakrishnan, G. Kathirvel, and A. Shaikh, "On the resiliency of virtual network functions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 152–157, Jul. 2017.
[11] F. He, T. Sato, and E. Oki, "Optimization model for backup resources allocation in middleboxes with importance," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1742–1755, Aug. 2019.
[12] Hewlett-Packard development company, "L.P. Servers for enterprise bladeSystem, rack & tower and hyperscale," 2015. [Online]. Available: http://www8.hp.com/us/en/products/servers/
[13] M. Huang, W. Liang, X. Shen, Y. Ma, and H. Kan, "Reliability-aware virtualized network function services provisioning in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2699–2713, Nov. 2020.
[14] M. Jia, W. Liang, and Z. Xu, "QoS-aware task offloading in distributed cloudlets with virtual network function services," in *Proc. ACM Int. Conf. Model., Anal. Simul. Wirel. Mobile Syst.*, 2017, pp. 109–116.
[15] J. Li, W. Liang, M. Huang, and X. Jia, "Providing reliability-aware virtualized network function services for mobile edge computing," in *Proc. Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 732–741.
[16] J. Li, W. Liang, M. Huang, and X. Jia, "Reliability-aware network service provisioning in mobile edge-cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 7, pp. 1545–1558, Jul. 2020.
[17] J. Li, W. Liang, and Y. Ma, "Robust service provisioning with service function chain requirements in mobile edge computing," *IEEE Trans. Netw. Serv. Manage.*, early access, Mar. 05, 2021, doi: 10.1109/TNSM.2021.3062650.

[18] W. Liang, Y. Ma, W. Xu, X. Jia, and S. Chau, “Reliability augmentation of requests with service function chain requirements in mobile edge-cloud networks,” in *Proc. Int. Conf. Parallel Process.*, 2020, pp. 1–11.
[19] S. Lin, W. Liang, and J. Li, “Reliability-aware service function chain provisioning in mobile edge-cloud networks,” in *Proc. Int. Conf. Comput. Commun. Netw.*, 2020, pp. 1–9.
[20] Y. Ma, W. Liang, J. Wu, and Z. Xu, “Throughput maximization of NFV-enabled multicasting in mobile edge cloud networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 2, pp. 394–407, Feb. 2020.
[21] Y. Mao, C. You, J. Zhang, K, Huang, and K. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Commun. Surv. Tut.*, vol. 19, pp. 2322–2358, Oct.–Dec. 2017.
[22] R. M. Nauss, “Solving the generalized assignment problem: An optimizing and heuristic approach,” *INFORMS J. Comput.*, vol. 15, no. 3, pp. 249–266, Aug. 2003.
[23] L. Qu, C. Assi, K. Shaban, and M. J. Khannaz, “A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks,” *IEEE Trans. Netw. Serv. Manage.*, vol. 14, no. 3, pp. 554–568, Sep. 2017.
[24] L. Qu, K. Shaban, and A. Assi, “Reliability-aware network service chaining in carrier-grade softwarized networks,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 558–573, Mar. 2018.
[25] P. Raghavan and C. D. Thompson, “Randomized rounding: A technique for provably good algorithms and algorithmic proofs,” *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.
[26] B. G. Rodriguez-Santana , A. M. Viveros, B. E. Carvajal-Gamez , and D. C. Trejo-Osorio , “Mobile computation offloading architecture for mobile augmented reality, case study: Visualization of cetacean skeleton,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 1, no. 7, pp. 665–671, Jun. 2016.
[27] X. Shang, Y. Huang, Z. Liu, and Y. Yang, “Reducing the service function chain backup cost over the edge and cloud by a self-adapting scheme,” *Proc. IEEE Conf. Comput. Commun.*, early access, Jan. 01, 2021, doi: 10.1109/TMC.2020.3048885.
[28] Z. Xu, W. Liang, M. Jia, M. Huang, and G. Mao, “Task offloading with network function services in a mobile edge-cloud network,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2072–2685, Nov. 2019.
[29] S. Yang, F. Li, R. Yahyappour, and X. Fu, “Delay-sensitive and availability-aware virtual network function scheduling for NFV,” *IEEE Trans Serv. Comput.*, early access, Jul. 09, 2019, doi: 10.1109/TSC.2019.2927339.
[30] R. Yu, G. Xue, and X. Zhang, “QoS-aware and reliable traffic steering for service function chaining in mobile networks,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2522–2531, Nov. 2017.

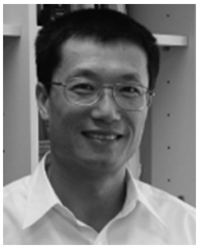
Weifa Liang (Senior Member, IEEE) received the BSc degree in computer science from Wuhan University, China, in 1984, the ME degree in computer science from the University of Science and Technology of China in 1989, and the PhD degree in computer science from the Australian National University in 1998. He is currently a professor at the Department of Computer Science, City University of Hong Kong, Hong Kong. Prior to that, he was a professor with the Research School of Computer Science, the Australian National University. His research interests include design and analysis of energy efficient routing protocols for wireless ad hoc and sensor networks, the Internet of Things, mobile edge computing, network function virtualization, software-defined networking, design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He is currently an associate editor for the *IEEE Transactions on Communications*.



Yu Ma received the BSc degree with (first class hons.) in computer science, in 2015, with the Australian National University, where he is currently working toward the PhD degree at the Research School of Computer Science. His research interests include software defined networking, Internet of Things, and social networking.

Wenzheng Xu (Member, IEEE) received the BSc, ME, and PhD degrees in computer science from Sun Yat-Sen University, Guangzhou, China, in 2008, 2010, and 2015, respectively. He is currently an associate professor at Sichuan University. He was a visitor with the Australian National University and The Chinese University of Hong Kong. His research interests include wireless ad hoc and sensor networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.

Zichuan Xu (Member, IEEE) received the BSc and ME degrees in computer science from the Dalian University of Technology in China, in 2008 and 2011, respectively, and the PhD degree in computer science from the Australian National University in 2016. He was a research associate with the Department of Electronic and Electrical Engineering, University College London, UK. He is currently an associate professor at the School of Software, Dalian University of Technology, China. His research interests include cloud computing, software-defined networking, network function virtualization, wireless sensor networks, routing protocol design for wireless networks, algorithmic game theory, and optimization problems.



Xiaohua Jia (Fellow, IEEE) received the BSc and MEng degrees from the University of Science and Technology of China, in 1984 and 1987, respectively, and the DSc degree in information science from the University of Tokyo in 1991. He is currently a chair professor at the Department of Computer Science, City University of Hong Kong. His research interests include cloud computing and distributed systems, computer networks, wireless sensor networks, and mobile wireless networks. From 2006 to 2009, he was an editor of the *IEEE Transactions on Parallel and Distributed Systems* and the *Journal of World Wide Web*. He is the general chair of the ACM MobiHoc 2008, the TPC co-chair of the IEEE MASS 2009, an area-chair of the IEEE INFOCOM 2010, the TPC co-chair of the IEEE GlobeCom 2010, the Ad Hoc and sensor networking symposium, and the panel co-chair of the IEEE INFOCOM 2011.

Wanlei Zhou (Senior Member, IEEE) received the BEng and MEng degrees in computer science and engineering from the Harbin Institute of Technology, Harbin, China, in 1982 and 1984, respectively, the PhD degree in computer science and engineering from The Australian National University, Canberra, Australia, in 1991, and the DSc degree (a higher Doctorate degree) from Deakin University in 2002. He has authored or coauthored more than 400 papers in refereed international journals and refereed international conferences proceedings, including many articles in the IEEE transactions and journals. His research interests include security, privacy, and distributed computing. He is currently the vice rector (academic affairs) and the dean of the Institute of Data Science, City University of Macau, China. Before joining the City University of Macau, he held various positions, including the head of the School of Computer Science, University of Technology Sydney, Australia, the Alfred Deakin professor, chair of information technology, an associate dean, and the head of the School of Information Technology with Deakin University, Australia.